Using a Flash ROM with the 21555 Non-Transparent PCI-to-PCI Bridge

Application Note

June 2001

Order Number: 278384-001

Contents

1.0 Introduction....1

2.0 Hardware Description....1

2.1 Supported Flash ROMs....1
2.2 Hardware Configuration....1
2.3 Timing....2

3.0 Flash ROM Access....3

3.1 Flash ROM Read by CSR Access....3
3.2 Flash ROM Write by CSR Access....5
3.3 Flash ROM Read by PCI Access....6

4.0 Multiple Device Attachment....6

4.1 Attaching Multiple Devices....7
4.2 Hardware Configuration....7

Figures

1 Flash ROM Hardware Configuration....2
2 Read and Write Strobe Timing....3
3 Parallel ROM Read Timing....4
4 Parallel ROM Write Timing....6
5 Multiple Device Attachment on ROM Interface....8

Tables

1 Flash ROM Interface Signals....2

1.0 Introduction

The 21555 supports the attachment of standard parallel 8-bit Flash ROM. The PCI expansion ROM base address register (BAR), at offset 30h, is offered with ROM sizes ranging between 4KB and 16MB. If required, this feature can provide the PCI expansion ROM interface for the subsystem. If the subsystem does not need this capability, the PCI expansion ROM BAR can be disabled by clearing bit 0 of this register.

After configuration of the 21555's primary PCI interface, the parallel ROM can be accessed by the host in the memory address range assigned by the PCI expansion ROM BAR. The Flash ROM is not directly accessible from the memory address space of the secondary PCI interface. However, it can be accessed indirectly from both the primary and secondary PCI interface using the 21555's command and status registers (CSRs).

2.0 Hardware Description

This section describes the supported Flash ROMs, hardware configuration, and timing.

2.1 Supported Flash ROMs

Flash ROMs compatible with Intel's 28F00x architecture can be used with the 21555 ROM interface. The clock input to the parallel ROM is the primary PCI clock, maximum frequency of 33 MHz, divided by two or 66 MHz divided by four, depending on the state of the P_M66ENA. The duty cycle of the clock is approximately 50%.

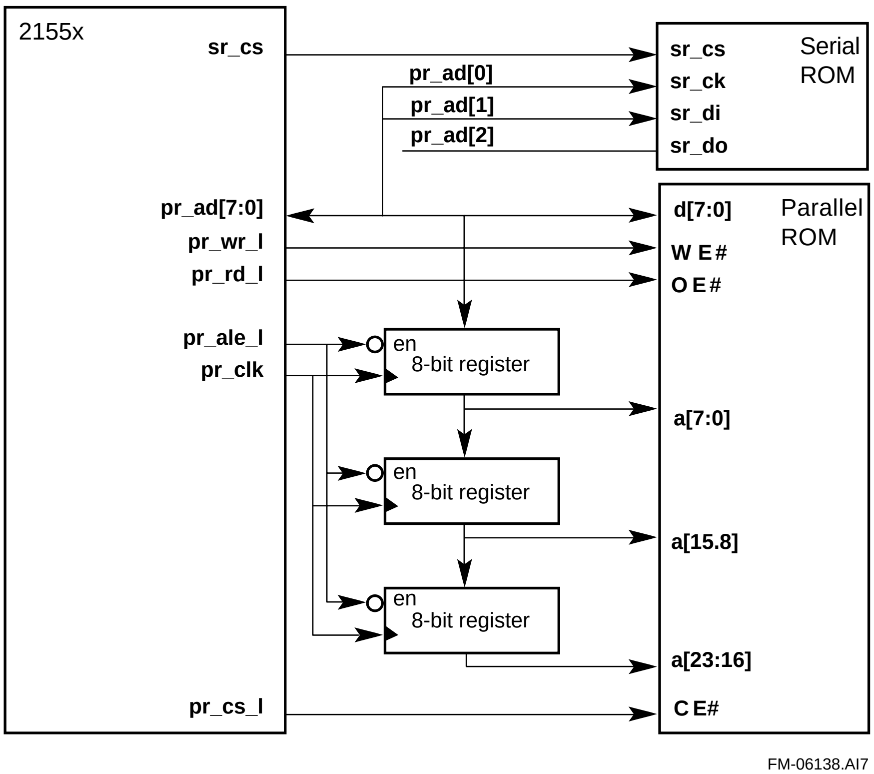
2.2 Hardware Configuration

The Flash ROM interface consists of thirteen signals as shown in the Table 1. Eight of these signals, **pr_ad[7:0]**, are multiplexed pins that act as both address and data lines. **pr_ad[2:0]** are shared with the serial ROM interface. Figure 1 shows the hardware set up for the Flash ROM interface. The ROM address is driven onto the eight **pr_ad** lines in three consecutive cycles. External octal D registers (377 D-type flip-flops) with active low enables capture the ROM address. The figure illustrates the connection of 16MB of ROM. If a smaller ROM is used, the address registers corresponding to the upper address bits can be removed.

Table 1. Flash ROM Interface Signals

Name	Type	Description
pr_ad[7:0]	Tristate bidirectional	Multiplexed address and data lines
pr_rd_l	Output	ROM output enable
pr_wr_l	Output	ROM write enable
pr_cs_l	Output	ROM chip select
pr_ale_l	Output	Address register clock enable
pr_clk	Output	Address register clock, **p_clk** divided by 2

Figure 1. Flash ROM Hardware Configuration



2.3 Timing

The ROM setup register (byte offsets 0C9h:0C8h) controls both the access time and read and write strobe timing. The access time specifies the number of **p_clk** cycles that the 21555 asserts **pr_cs_l**. It is specified as a multiple of the **p_clk** and must be set to 8, 16, 64, or 256 times the length of a **p_clk** cycle. The reset value is 8.

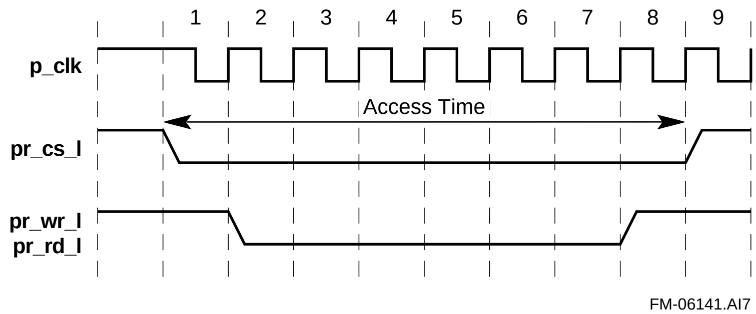
The read and write strobe timing is defined by an 8-bit field in the ROM setup register [15:8]. During the device select assertion, the setup time, duration, and hold time of **pr_rd_l** and **pr_wr_l** are specified by these bits. Each bit corresponds to 1/8 of the access time, which translates to 1, 2, 8, or 32 **p_clk** cycles. Bit 8 in the register corresponds to the first cycle in time and bit 15 to the last

cycle. If a bit is 0, the read or write strobe is deasserted. When a bit is 1, the read or write strobe is asserted. **pr_cs_l** is asserted at the beginning of a cycle and deasserted at the end of the last cycle. The reset value for the ROM setup register [15:8] is 0111 1110b. Since the reset value of the access time is eight **p_clk** cycles (ROM setup register [1:0] equal 00), a read or write access would have the following timing:

- Cycle 1 asserts **pr_cs_l**
- Cycles 2 through 7 asserts both **pr_cs_l,** and either **pr_rd_l** or **pr_wr_l**
- Cycle 8 deasserts **pr_rd_l** or **pr_wr_l**
- Cycle 9 deasserts **pr_cs_l**

Figure 2 shows the read and write strobe timing.

Figure 2. Read and Write Strobe Timing



3.0 Flash ROM Access

This section describes Flash ROM read and write cycles by CSR access. It also describes Flash ROM reads through PCI access.

3.1 Flash ROM Read by CSR Access

Flash ROM byte reads can be performed by CSR access of the following registers:

- ROM control (offset 0CFh)
- ROM address (offset 0CCh)
- ROM data (offset 0CAh)

The following sequence describes a byte read access:

1. Write the byte address offset to the ROM address register.
2. Write the parallel ROM start bit [1] to a 1, the serial ROM start bit [0] to a 0, and the read/write control bit [2] to a 0 in the ROM control register. This can be done in one CSR access.
3. Poll the parallel ROM start bit. When it is a 0, the operation is complete and the data can be obtained from the ROM data register.
4. Read the contents of the ROM data register.

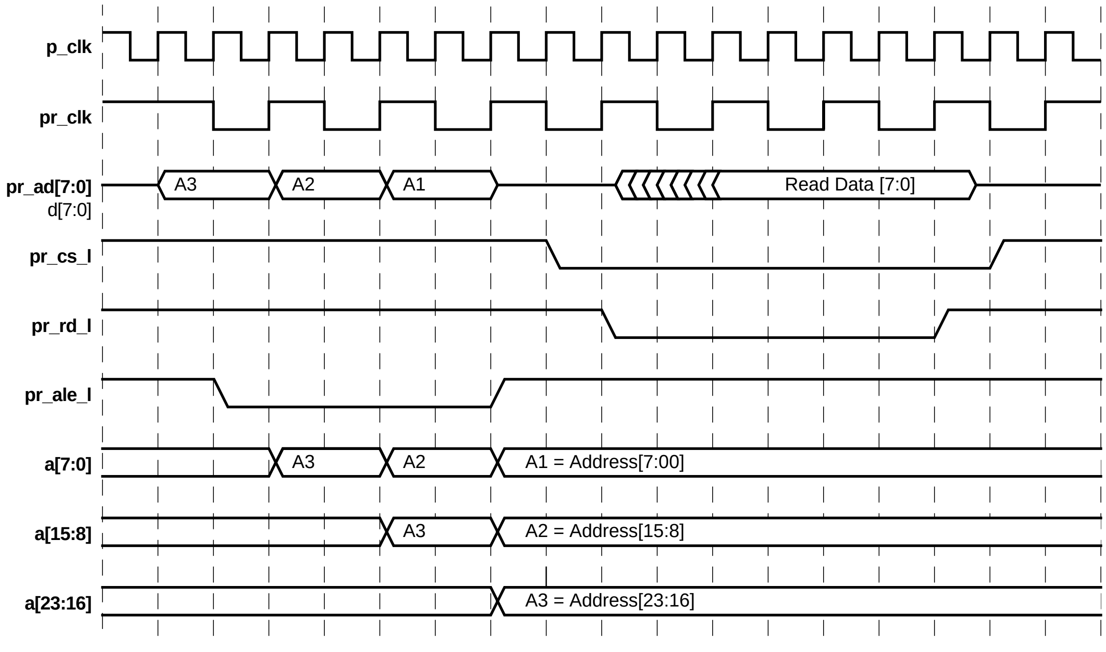
Example code for a ROM read through CSR access is provided as follows:

```
/* put the interested ROM address into ROM address register */

PUT32(CSR_BASE+0xCC, byteAddressToRead);

/* start the transaction from 21555 to the Flash ROM by setting start bit */

PUT8(CSR_BASE+0xCF,ParallelStartBusy);

/* wait for busy bit to clear */

while (GET8(CSR_BASE+0xCF) & ParallelStartBusy);

/* read the byte of data from the ROM */

value = GET8(CSR_BASE+0xCA);
```

Figure 3 shows the timing diagram for the parallel ROM read operation. The 21555 drives address bits [23:16] on the **pr_ad[7:0]** pins, asserts **pr_ale_l** to enable the address registers, and drives **pr_clk** high to latch the address in the first external register. While **pr_ale_l** remains low, address bits [15:8] are driven on the **pr_ad** pins. **pr_clk** is driven high again, to latch address bits [15:8] in the first external register and address bits [23:16] are latched into the second external register. The process is repeated a third time with address bits [7:0] driven on the **pr_ad** pins.

When **pr_clk** is driven high again, address bits [7:0] are latched into the first external register, address bits [15:8] into the second external register, and address bits [23:16] into the third external register. The 21555 then deasserts **pr_ale_l** and asserts the **pr_cs_l** and **pr_rd_l** pins according to the strobe setup timing specified in the strobe mask of the ROM setup register (byte offsets 0C9, 0C8h). The parallel ROM drives the data onto **pr_ad[7:0**] and the 21555 samples the read data and deasserts both the **pr_rd_l** and **pr_cs_l** pins according to the times specified in the ROM setup register. The 21555 then clears the parallel ROM start bit and valid data can now be read from the ROM data register

Figure 3. Parallel ROM Read Timing



3.2 Flash ROM Write by CSR Access

Flash ROM byte writes can be performed by CSR access of the following registers:

- ROM control (offset 0CFh)
- ROM address (offset 0CCh)
- ROM data (offset 0CAh)

The following sequence describes a byte read access:

1. Write the byte address offset to the ROM address register.
2. Write one byte of data into the ROM data register.
3. Write the parallel ROM start bit [1] to a 1, the serial ROM start bit [0] to a 0, and the read/write control bit [2] to a 1 in the ROM control register. This can be done in one CSR access.
4. The access is complete when the parallel ROM start bit in the ROM control register is a 0.

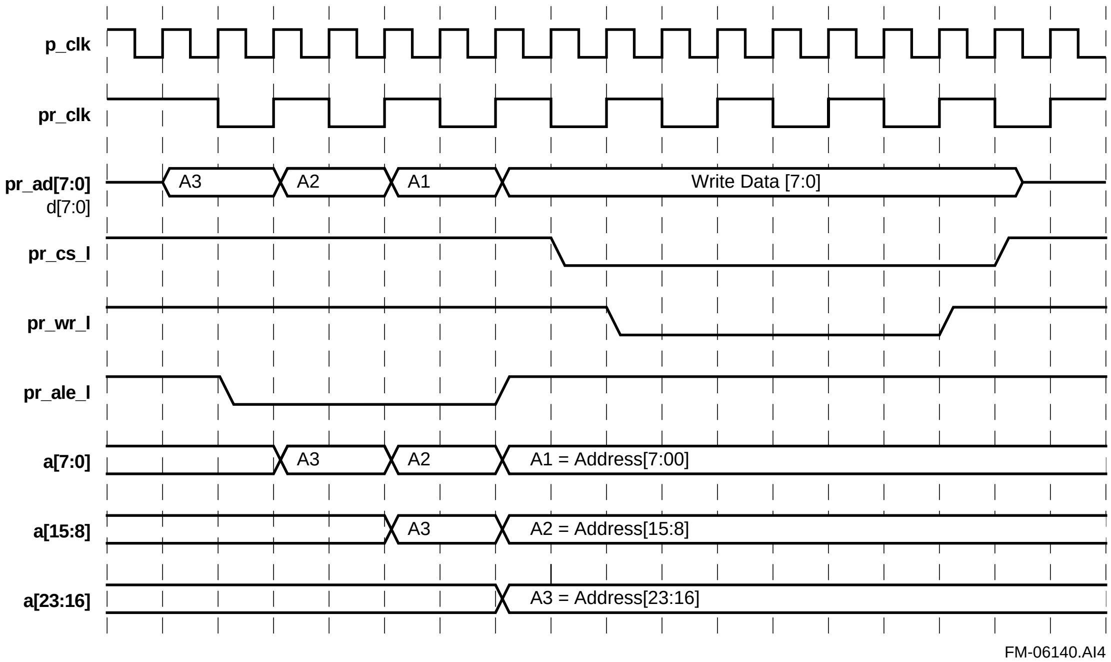
Example code for a ROM write through CSR access is provided as follows:

```
/* put the interested ROM address into ROM address register */

PUT32(CSR_BASE+0xCC, byteAddressToWrite);

/* start the transaction from 21555 to the Flash ROM by setting start bit */

/* Set the write command bit also. The absence of this bit is a read command */

PUT8(CSR_BASE+0xCF,ParallelStartBusy|ParallelWrite);

/* wait for busy bit to clear */

while (GET8(CSR_BASE+0xCF) & ParallelStartBusy);
```

Figure 4 shows the timing diagram for the parallel ROM write operation. The 21555 drives address bits [23:16] on the **pr_ad[7:0]** pins, asserts **pr_ale_l** to enable the address registers, and drives **pr_clk** high to latch the address in the first external register. While **pr_ale_l** remains low, address bits [15:8] are driven on the **pr_ad** pins. **pr_clk** is driven high again, to latch address bits [15:8] in the first external register and address bits [23:16] are latched into the second external register. The process is repeated a third time with address bits [7:0] driven on the **pr_ad** pins.

When **pr_clk** is driven high again, address bits [7:0] are latched into the first external register, address bits [15:8] into the second external register, and address bits [23:16] into the third external register. The 21555 then deasserts **pr_ale_l** and drives the data onto **pr_ad[7:0]**. The 21555 then asserts the **pr_cs_l** and **pr_wr_l** pins according to the strobe setup timing specified in the strobe mask of the ROM setup register (byte offsets 0C9h:0C8h). These signals are subsequently deasserted according to the times specified in the ROM setup register. The 21555 then clears the parallel ROM start bit in the ROM control register.

Figure 4. Parallel ROM Write Timing



3.3 Flash ROM Read by PCI Access

A Flash ROM Dword read is automatically performed when a PCI read on the 21555's primary bus has an address within the address range defined by the primary expansion ROM BAR. The 21555 treats a memory read through the primary expansion ROM BAR as a delayed read and returns a target retry. The 21555 then performs four consecutive byte reads of the ROM and when complete, it returns the read data on the next read attempt to the delayed transaction address. The parallel ROM start/busy bit is automatically set when the read is initiated and is cleared when the Flash ROM read is complete.

Note: The CSR access method should not be used for the parallel ROM when reads to the parallel ROM through the primary expansion ROM BAR are taking place.

4.0 Multiple Device Attachment

This section describes attaching multiple devices to the parallel ROM port and the external hardware that is required.

4.1 Attaching Multiple Devices

The 21555 allows additional devices to be attached to the parallel ROM interface. By setting the multiple device enable bit (9) of the chip control 0 register (offsets: CDh:CCh), two ROM interface signals (**pr_ale_l** and **pr_cs_l**) are slightly redefined to support multiple devices. In this mode, the maximum ROM size is reduced because the upper address lines are used to decode device select lines.

4.2 Hardware Configuration

For this mode of operation, an external decoder (138 type 1-of-8 decoder/demultiplexer) is required as shown in Figure 5. The decoder decodes the upper address bits into device select lines. The **pr_ale_l** signal is used to enable the device select decoder in addition to enabling the clocking of the address registers. When **pr_ale_l** is low, the address registers are enabled. When **pr_ale_l** is high, the device select decoder is enabled.

In the multiple device mode, the **pr_cs_l** output is redefined to be a device ready input (**pr_rdy**). When using multiple device mode and multiple devices are attached, **pr_cs_l** should be tied high through an external 5 KΩ resistor. When **pr_rdy** is deasserted, the device select line and read or write strobe assertions are extended until **pr_rdy** is asserted again. The read and write strobes are deasserted upon detection of chip select deassertion (falling **pr_ale_l**). The hold time is specified by the strobe mask.

Figure 5. Multiple Device Attachment on ROM Interface